



NetApp儲存整合概述

NetApp Solutions

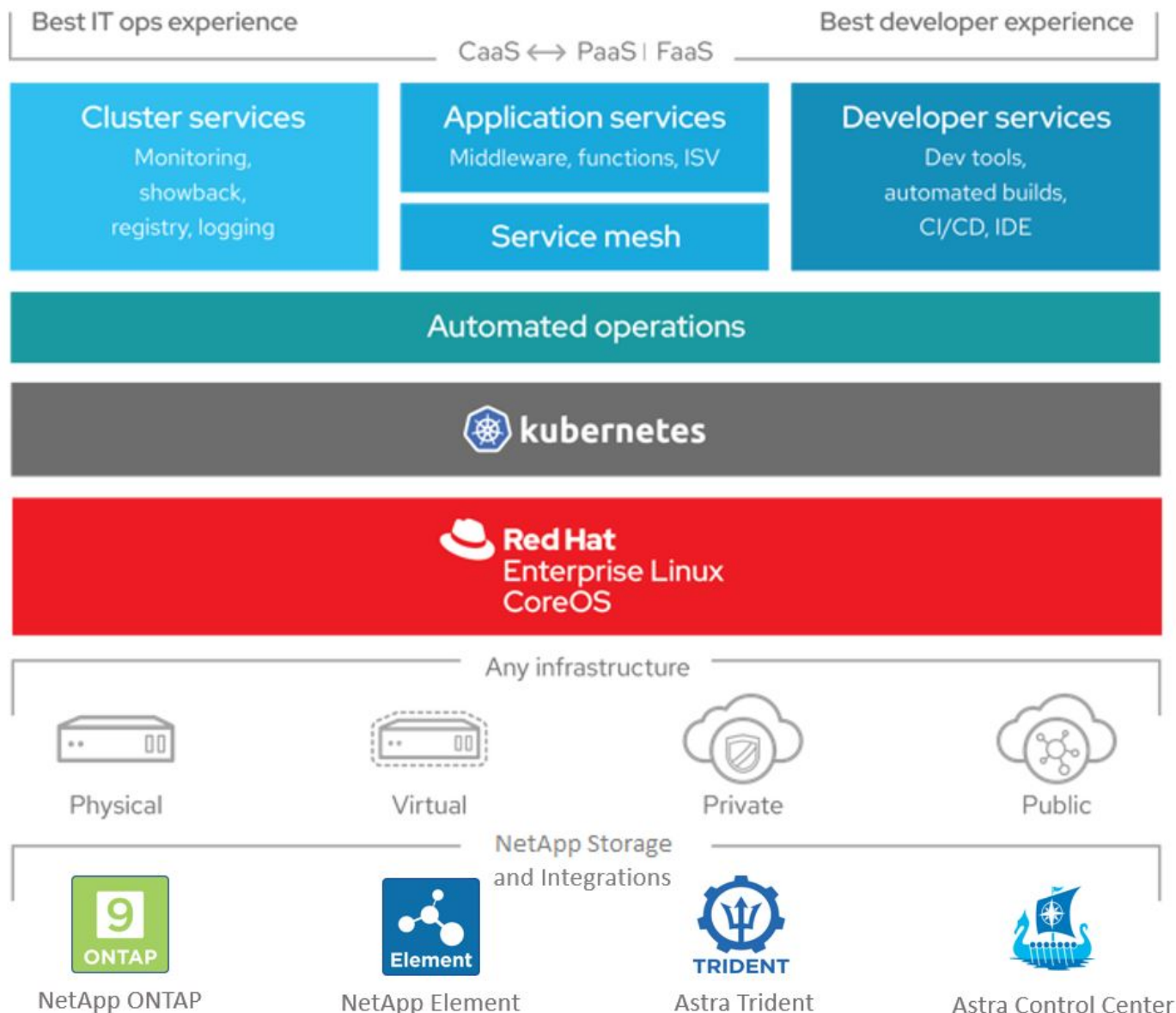
NetApp
April 12, 2024

目錄

NetApp儲存整合概述	1
NetApp Astra Control Center總覽	2
Astra Trident總覽	30

NetApp儲存整合概述

NetApp提供多種產品來協助您協調及管理以容器為基礎的環境中的持續資料、例如Red Hat OpenShift。



NetApp Astra Control提供豐富的儲存設備與應用程式感知資料管理服務、適用於狀態明確的Kubernetes工作負載、採用NetApp資料保護技術。Astra Control Service可支援雲端原生Kubernetes部署中的狀態工作負載。Astra Control Center可支援內部部署中的狀態工作負載、例如Red Hat OpenShift。如需詳細資訊、請參閱NetApp Astra Control網站 ["請按這裡"](#)。

NetApp Astra Trident是開放原始碼且完全支援的儲存協調工具、適用於容器和Kubernetes配送、包括Red Hat OpenShift。如需詳細資訊、請造訪Astra Trident網站 ["請按這裡"](#)。

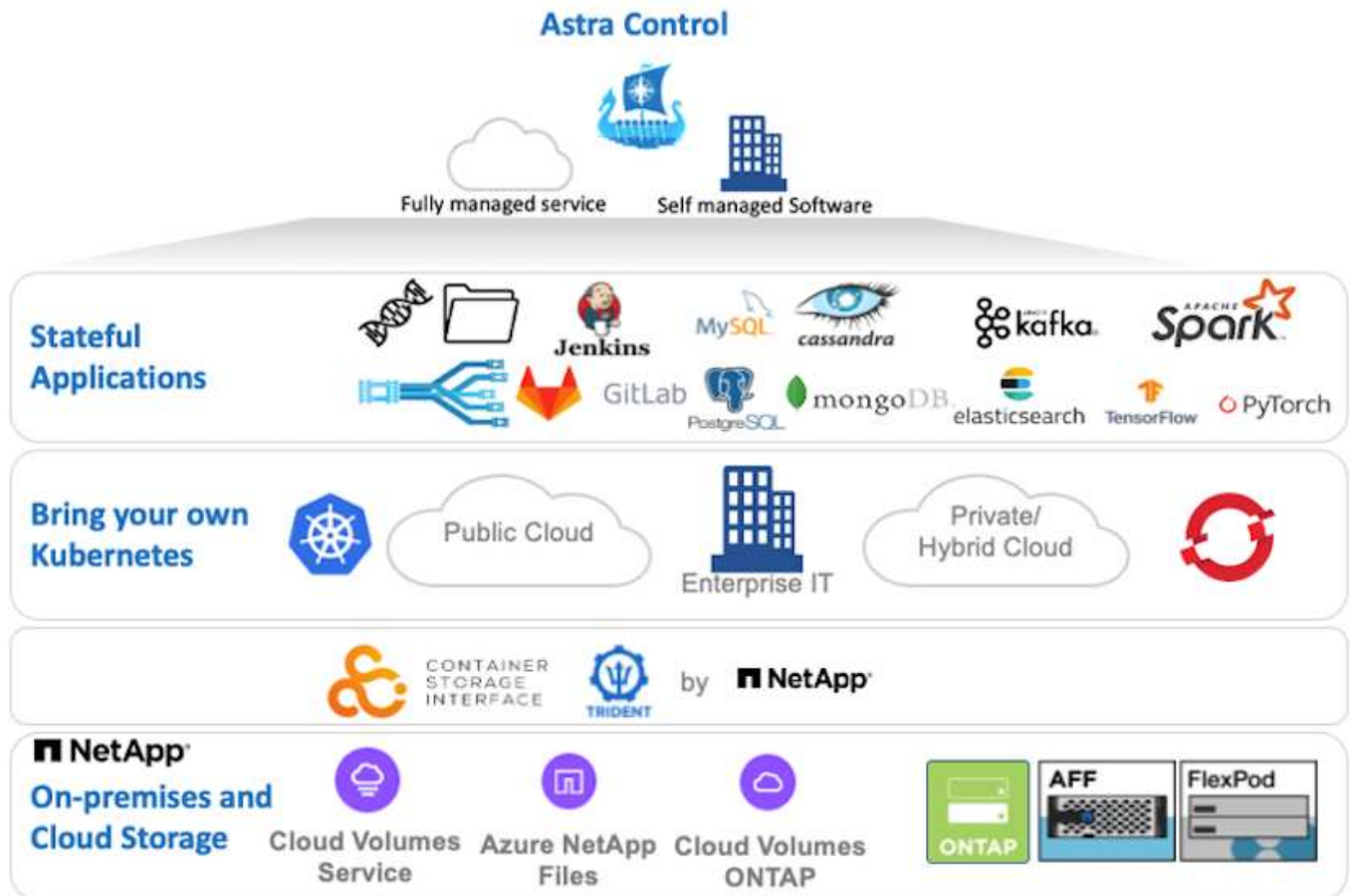
以下頁面提供更多有關NetApp產品的資訊、這些產品已通過Red Hat OpenShift with NetApp解決方案的應用程式與持續儲存管理驗證：

- ["NetApp Astra控制中心"](#)

- ["NetApp Astra Trident"](#)

NetApp Astra Control Center總覽

NetApp Astra Control Center提供豐富的儲存設備與應用程式感知資料管理服務、適用於部署在內部部署環境中且採用NetApp資料保護技術的狀態式Kubernetes工作負載。



NetApp Astra Control Center可安裝在Red Hat OpenShift叢集上、該叢集已部署Astra Trident儲存Orchestrator、並已設定儲存類別和儲存後端、以供NetApp ONTAP orchstorage系統使用。

如需安裝及組態Astra Trident以支援Astra Control Center、請參閱 ["本文檔"](#)。

在雲端連線的環境中、Astra Control Center會使用Cloud Insights 效益技術來提供進階監控和遙測功能。如果沒有支援功能、則可透過開放式指標端點、將有限的監控和遙測（7天的數據價值）匯出至Kubernetes原生監控工具（Prometheus和Grafana） Cloud Insights。

Astra Control Center已完全整合至NetApp AutoSupport 的整套功能和Active IQ 功能、可為使用者提供支援、協助疑難排解、以及顯示使用統計資料。

除了Astra Control Center的付費版本、我們也提供90天的評估授權。評估版本可透過電子郵件和社群（Slack通路）獲得支援。客戶可以存取這些和其他知識庫文章、以及產品內建支援儀表板所提供的文件。

若要開始使用NetApp Astra Control Center、請造訪 ["Astra網站"](#)。

Astra Control Center安裝先決條件

1. 一個或多個Red Hat OpenShift叢集。目前支援版本4.6 EUS和4.7。
2. 每個Red Hat OpenShift叢集上都必須已安裝並設定Astra Trident。
3. 一或多ONTAP 個執行ONTAP 不穩定9.5或更新版本的NetApp不穩定儲存系統。



最佳實務做法是在站台上安裝每個OpenShift、以擁有專屬的SVM來進行持續儲存。多站台部署需要額外的儲存系統。

4. 每個OpenShift叢集上都必須設定Trident儲存後端、並以SVM作為後盾ONTAP、以供支援整個叢集。
5. 在每個OpenShift叢集上設定的預設StorageClass、以Astra Trident做為儲存資源配置程式。
6. 必須在每個OpenShift叢集上安裝及設定負載平衡器、才能平衡負載並公開OpenShift Services。



請參閱連結 "[請按這裡](#)" 以取得已針對此目的驗證的負載平衡器相關資訊。

7. 必須設定私有映像登錄來裝載NetApp Astra Control Center映像。



請參閱連結 "[請按這裡](#)" 若要安裝及設定OpenShift Private登錄以供此用途。

8. 您必須擁有Red Hat OpenShift叢集的叢集管理存取權。
9. 您必須擁有NetApp ONTAP 等群集的管理員存取權。
10. 安裝了Docker或podman、tridentctl、及occ或kubectl工具的管理工作站、並新增至\$PATH。



Docker安裝必須具有大於20.10的Docker版本、而Podman安裝的podman版本必須大於3.0。

安裝Astra Control Center

使用作業系統集線器

1. 登入NetApp支援網站、下載最新版本的NetApp Astra Control Center。若要這麼做、必須附上NetApp帳戶的授權。下載後、將其傳輸至管理工作站。



若要開始使用Astra Control試用授權、請造訪 "[Astra註冊網站](#)"。

2. 解壓縮tar ball並將工作目錄變更為所產生的資料夾。

```
[netapp-user@rhel7 ~]$ tar -vxzf astra-control-center-  
21.12.60.tar.gz  
[netapp-user@rhel7 ~]$ cd astra-control-center-21.12.60
```

3. 開始安裝之前、請先將Astra Control Center映像推送到映像登錄。您可以選擇使用Docker或Podman來執行這項作業、本步驟提供這兩者的說明。

Podman

- a. 將登錄FQDN與組織/命名空間/專案名稱匯出為環境變數「正式作業」。

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

- b. 登入登錄。

```
[netapp-user@rhel7 ~]$ podman login -u ocp-user -p password --tls-verify=false astra-registry.apps.ocp-vmw.cie.netapp.com
```



如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖、而非密碼「podman登入-u ocp-user-p權杖—tlS-VERIF=假astra-registry.apps.ocp-vmw.cie.netapp.com」。



或者、您也可以建立服務帳戶、指派登錄編輯器和/或登錄檢視器角色（取決於您是否需要推入/拉取存取）、然後使用服務帳戶的權杖登入登錄。

- c. 建立Shell指令碼檔案、然後將下列內容貼入其中。

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
    # Load to local cache. And store the name of the loaded
    image trimming the 'Loaded images: '
    astraImage=$(podman load --input ${astraImageFile} | sed
's/Loaded image(s): //' )
    astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
    # Tag with local image repo.
    podman tag ${astraImage} ${REGISTRY}/${astraImage}
    # Push to the local repo.
    podman push ${REGISTRY}/${astraImage}
done
```



如果您的登錄使用不受信任的憑證、請編輯Shell指令碼、並針對podman推送命令「podman push \$註冊表/\$ (ECAECA\$astraImage | sed 's/\]/\]/+') -TLS-VERIFY=假」使用「-TLS-VERIFY」。

- d. 將檔案設定為可執行檔。

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

e. 執行Shell指令碼。

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```


Docker

- a. 將登錄FQDN與組織/命名空間/專案名稱匯出為環境變數「正式作業」。

```
[netapp-user@rhel7 ~]$ export REGISTRY=astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra
```

- b. 登入登錄。

```
[netapp-user@rhel7 ~]$ docker login -u ocp-user -p password astra-registry.apps.ocp-vmw.cie.netapp.com
```



如果您使用「kubeadmin」使用者登入私有登錄、請使用權杖而非密碼-「docker login-u ocp-user-p權杖astra-registry.apps.ocp-vmw.cie.netapp.com」。



或者、您也可以建立服務帳戶、指派登錄編輯器和/或登錄檢視器角色（取決於您是否需要推入/拉取存取）、然後使用服務帳戶的權杖登入登錄。

- c. 建立Shell指令碼檔案、然後將下列內容貼入其中。

```
[netapp-user@rhel7 ~]$ vi push-images-to-registry.sh

for astraImageFile in $(ls images/*.tar) ; do
    # Load to local cache. And store the name of the loaded
    image trimming the 'Loaded images: '
    astraImage=$(docker load --input ${astraImageFile} | sed
's/Loaded image: //' )
    astraImage=$(echo ${astraImage} | sed 's!localhost/!!')
    # Tag with local image repo.
    docker tag ${astraImage} ${REGISTRY}/${astraImage}
    # Push to the local repo.
    docker push ${REGISTRY}/${astraImage}
done
```

- d. 將檔案設定為可執行檔。

```
[netapp-user@rhel7 ~]$ chmod +x push-images-to-registry.sh
```

- e. 執行Shell指令碼。

```
[netapp-user@rhel7 ~]$ ./push-images-to-registry.sh
```

4. 使用非公開信任的私有映像登錄時、請將映像登錄TLS憑證上傳至OpenShift節點。若要這麼做、請使用TLS憑證在openshift-config命名空間中建立組態對應、並將其修補至叢集映像組態、使憑證成為信任的憑證。

```
[netapp-user@rhel7 ~]$ oc create configmap default-ingress-ca -n  
openshift-config --from-file=astra-registry.apps.ocp  
-vmw.cie.netapp.com=tls.crt  
  
[netapp-user@rhel7 ~]$ oc patch image.config.openshift.io/cluster  
--patch '{"spec":{"additionalTrustedCA":{"name":"default-ingress-  
ca"}}}' --type=merge
```



如果您使用OpenShift內部登錄搭配來自入口操作員的預設TLS憑證搭配路由、您仍需依照前一個步驟將憑證修補成路由主機名稱。若要從入口操作員擷取憑證、您可以使用命令「`oc extract secret /路由器-ca --keys=ls.crt -n openshift-inet-opoperators`」。

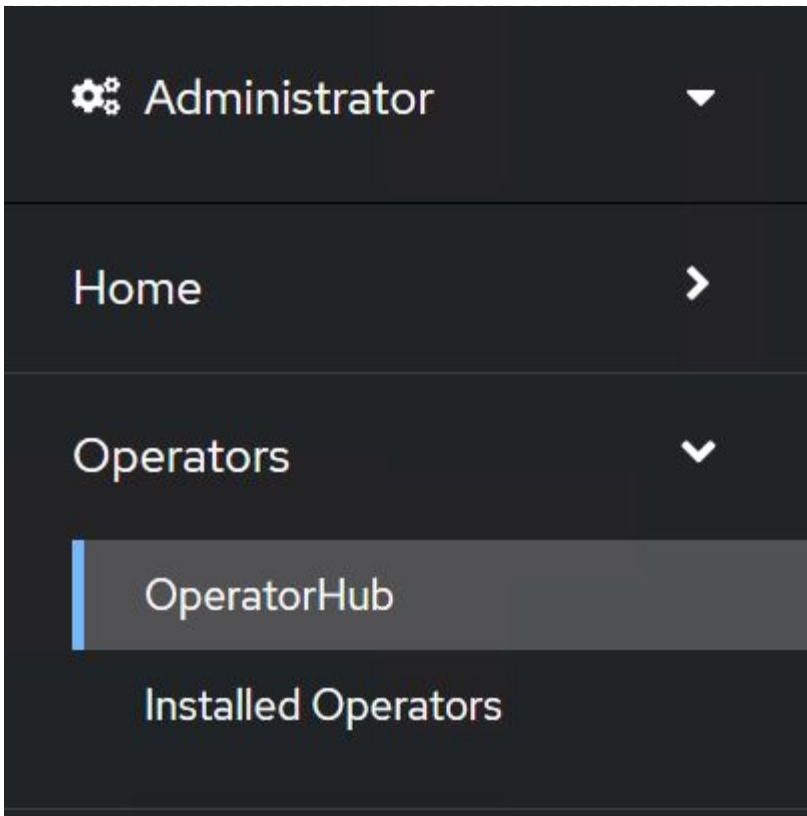
5. 為Astra Control Center建立命名空間「NetApp-acc operator」。

```
[netapp-user@rhel7 ~]$ oc create ns netapp-acc-operator  
  
namespace/netapp-acc-operator created
```


6. 使用認證資料建立秘密、以登入「NetApp-acc operator」命名空間中的映像登錄。

```
[netapp-user@rhel7 ~]$ oc create secret docker-registry astra-  
registry-cred --docker-server=astra-registry.apps.ocp  
-vmw.cie.netapp.com --docker-username=ocp-user --docker  
-password=password -n netapp-acc-operator  
  
secret/astra-registry-cred created
```

7. 使用叢集管理存取權登入Red Hat OpenShift GUI主控台。
8. 從Perspective（透視）下拉列表中選擇Administrator（管理員
9. 瀏覽至「運算子」>「運算子中樞」、然後搜尋Astra。



10. 選取「NetApp-acc operator」方塊、然後按一下「Install（安裝）」。



netapp-acc-operator
21.12.63-1 provided by NetApp

✕

Install

Latest version
21.12.63-1

Capability level
☒ Basic Install
☐ Seamless Upgrades
☐ Full Lifecycle
☐ Deep Insights
☐ Auto Pilot

Provider type
Certified

Provider
NetApp

Astra Control is an application-aware data management solution that manages, protects and moves data-rich Kubernetes workloads in both public clouds and on-premises.

Astra Control enables data protection, disaster recovery, and migration for your Kubernetes workloads, leveraging NetApp's industry-leading data management technology for snapshots, backups, replication and cloning.

How to deploy Astra Control

Refer to [Installation Procedure](#) to deploy Astra Control Center using the Operator.

Documentation

Refer to [Astra Control Center Documentation](#) to complete the setup and start managing applications.

11. 在Install Operator（安裝操作員）畫面上、接受所有預設參數、然後按一下「Install（安裝）」。

Install Operator

Install your Operator by subscribing to one of the update channels to keep the Operator up to date. The strategy determines either manual or automatic updates.

Update channel *

- ☐ alpha
- ☒ stable

Installation mode *

- ☒ All namespaces on the cluster (default)
Operator will be available in all Namespaces.
- ☐ A specific namespace on the cluster
This mode is not supported by this Operator

Installed Namespace *

PR netapp-acc-operator (Operator recommended)

⚠ Namespace already exists

Namespace **netapp-acc-operator** already exists and will be used. Other users can already have access to this namespace.

Approval strategy *

- ☒ Automatic
- ☐ Manual

Install

Cancel

 **netapp-acc-operator**
provided by NetApp

Provided APIs

 **Astra Control Center**

AstraControlCenter is the Schema for the astracontrolcenters API

12. 等待操作員安裝完成。



netapp-acc-operator
21.12.63-1 provided by NetApp



Installing Operator

InstallWaiting: installing: waiting for deployment acc-operator-controller-manager to become ready: Waiting for rollout to finish: 0 of 1 updated replicas are available...

The Operator is being installed. This may take a few minutes.

[View installed Operators in Namespace netapp-acc-operator](#)

13. 一旦操作員安裝成功、請瀏覽至「View operator」（檢視操作員）。



netapp-acc-operator
21.12.63-1 provided by NetApp



Installed operator - ready for use

[View Operator](#)

[View installed Operators in Namespace netapp-acc-operator](#)

14. 然後按一下操作者中Astra Control Center的「Create Instance」（建立執行個體）。

[Installed Operators](#) > [Operator details](#)



netapp-acc-operator
21.12.63-1 provided by NetApp

[Details](#)

[YAML](#)

[Subscription](#)

[Events](#)

[Astra Control Center](#)

Provided APIs

ACC Astra Control Center

AstraControlCenter is the Schema for
the astracontrolcenters API

[+ Create instance](#)

15. 填寫「Create適用的」表單欄位、然後按一下「Create」（建立）。
- （可選）編輯Astra Control Center執行個體名稱。
 - （可選）啟用或停用自動支援。建議保留「自動支援」功能。
 - 輸入Astra Control Center的FQDN。
 - 輸入Astra Control Center版本；預設會顯示最新版本。

- e. 輸入Astra Control Center的帳戶名稱和管理員詳細資料、例如名字、姓氏和電子郵件地址。
- f. 輸入Volume回收原則、預設為保留。
- g. 在「Image登錄」中、輸入登錄的FQDN以及將映像推送到登錄時所提供的組織名稱（在此範例中為「astra-registry.apps.ocp-vmw.cie.netapp.com/netapp-astra`」）。
- h. 如果您使用需要驗證的登錄、請在「映像登錄」區段中輸入機密名稱。
- i. 設定Astra Control Center資源限制的擴充選項。
- j. 如果您要將PVCS放置在非預設儲存類別上、請輸入儲存類別名稱。
- k. 定義客戶需求日處理偏好設定。

Project: netapp-acc-operator ▼

Name *

astra

Labels

app=frontend

Account Name *

HCG Solutions Engineering

Astra Control Center account name

Astra Address *

astra-control-center.cie.netapp.com

AstraAddress defines how Astra will be found in the data center. This IP address and/or DNS A record must be created prior to provisioning Astra Control Center. Example - "astra.example.com" The A record and its IP address must be allocated prior to provisioning Astra Control Center

Astra Version *

2112.60

Version of AstraControlCenter to deploy. You are provided a Helm repository with a corresponding version. Example - 1.5.2, 1.4.2-patch

Email *

solutions_tme@netapp.com

EmailAddress will be notified by Astra as events warrant.

Auto Support * >

AutoSupport indicates willingness to participate in NetApp's proactive support application, NetApp Active IQ. The default election is true and indicates support data will be sent to NetApp. An empty or blank election is the same as a default election. Air gapped installations should enter false.

First Name

HCG

The first name of the SRE supporting Astra.

Last Name

The last name of the SRE supporting Astra.

Image Registry

The container image registry that is hosting the Astra application images, ACC Operator and ACC Helm Repository.

Name

The name of the image registry. For example "example.registry/astra". Do not prefix with protocol.

Secret

The name of the Kubernetes secret that will authenticate with the image registry.

Volume Reclaim Policy

Reclaim policy to be set for persistent volumes

Astra Resources Scaler

Scaling options for AstraControlCenter Resource limits.

Storage Class

The storage class to be used for PVCs. If not set, default storage class will be used.

Crds

Options for how ACC should handle CRDs.

自動[可執行]

1. 若要使用Ansible教戰手冊來部署Astra Control Center、您需要安裝Ansible的Ubuntu / RHEL機器。請依照程序進行 ["請按這裡"](#) 適用於 Ubuntu 和 RHEL 。
2. 複製裝載可執行內容的GitHub儲存庫。

```
git clone https://github.com/NetApp-
Automation/na_astra_control_suite.git
```

3. 登入NetApp支援網站、下載最新版的NetApp Astra Control Center。若要這麼做、必須附上NetApp帳戶的授權。下載後、將其傳輸至工作站。



若要開始使用Astra Control試用授權、請造訪 ["Astra註冊網站"](#)。

4. 建立或取得具有OpenShift叢集管理員存取權的Kbeconfig檔案、以安裝Astra Control Center。
5. 將目錄變更為na_astra_control_suite。

```
cd na_astra_control_suite
```

6. 編輯「vars/vars.yml」檔案、並在變數中填入所需資訊。

```
#Define whether or not to push the Astra Control Center images to
your private registry [Allowed values: yes, no]
push_images: yes

#The directory hosting the Astra Control Center installer
installer_directory: /home/admin/

#Specify the ingress type. Allowed values - "AccTraefik" or
"Generic"
#"AccTraefik" if you want the installer to create a LoadBalancer
type service to access ACC, requires MetallB or similar.
#"Generic" if you want to create or configure ingress controller
yourself, installer just creates a ClusterIP service for traefik.
ingress_type: "AccTraefik"

#Name of the Astra Control Center installer (Do not include the
extension, just the name)
astra_tar_ball_name: astra-control-center-22.04.0

#The complete path to the kubeconfig file of the
kubernetes/openshift cluster Astra Control Center needs to be
installed to.
hosting_k8s_cluster_kubeconfig_path: /home/admin/cluster-
kubeconfig.yml

#Namespace in which Astra Control Center is to be installed
astra_namespace: netapp-astra-cc

#Astra Control Center Resources Scaler. Leave it blank if you want
to accept the Default setting.
astra_resources_scaler: Default

#Storageclass to be used for Astra Control Center PVCs, it must be
created before running the playbook [Leave it blank if you want the
PVCs to use default storageclass]
astra_trident_storageclass: basic

#Reclaim Policy for Astra Control Center Persistent Volumes [Allowed
values: Retain, Delete]
storageclass_reclaim_policy: Retain
```



```

#Private Registry Details
astra_registry_name: "docker.io"

#Whether the private registry requires credentials [Allowed values:
yes, no]
require_reg_creds: yes

#If require_reg_creds is yes, then define the container image
registry credentials
#Usually, the registry namespace and usernames are same for
individual users
astra_registry_namespace: "registry-user"
astra_registry_username: "registry-user"
astra_registry_password: "password"

#Kuberenets/OpenShift secret name for Astra Control Center
#This name will be assigned to the K8s secret created by the
playbook
astra_registry_secret_name: "astra-registry-credentials"

#Astra Control Center FQDN
acc_fqdn_address: astra-control-center.cie.netapp.com

#Name of the Astra Control Center instance
acc_account_name: ACC Account Name

#Administrator details for Astra Control Center
admin_email_address: admin@example.com
admin_first_name: Admin
admin_last_name: Admin

```

7. 執行教戰手冊以部署Astra Control Center。本方針要求特定組態具備root權限。

如果執行方針的使用者是root或設定了無密碼Sudo、請執行下列命令來執行方針。

```
ansible-playbook install_acc_playbook.yml
```

如果使用者已設定以密碼為基礎的Sudo存取、請執行下列命令來執行方針、然後輸入Sudo密碼。

```
ansible-playbook install_acc_playbook.yml -K
```

安裝後步驟

1. 安裝可能需要幾分鐘的時間才能完成。確認「NetApp-Astra -cc」命名空間中的所有Pod和服務均已啟動並正在執行。

```
[netapp-user@rhel7 ~]$ oc get all -n netapp-astra-cc
```

2. 檢查「acc oper-manager-manager」記錄、確保安裝完成。

```
[netapp-user@rhel7 ~]$ oc logs deploy/acc-operator-controller-manager -n  
netapp-acc-operator -c manager -f
```



下列訊息表示Astra Control Center安裝成功。

```
{"level":"info","ts":1624054318.029971,"logger":"controllers.AstraControlCenter","msg":"Successfully Reconciled AstraControlCenter in [seconds]s","AstraControlCenter":"netapp-astra-cc/astra","ae.Version":"[21.12.60]"}
```

3. 登入Astra Control Center的使用者名稱是CRD檔案中所提供系統管理員的電子郵件地址、密碼是附加於Astra Control Center UUID的字串「ACC-」。執行下列命令：

```
[netapp-user@rhel7 ~]$ oc get astracontrolcenters -n netapp-astra-cc  
NAME      UUID  
astra     345c55a5-bf2e-21f0-84b8-b6f2bce5e95f
```



在此範例中、密碼為「ACC-345c55a5-bf2e-21f0-843b8-b6f2bce5e95f」。

4. 取得traefik服務負載平衡器IP。

```
[netapp-user@rhel7 ~]$ oc get svc -n netapp-astra-cc | egrep  
'EXTERNAL|traefik'
```

NAME	TYPE	CLUSTER-IP
EXTERNAL-IP	PORT(S)	
AGE		
traefik	LoadBalancer	172.30.99.142
10.61.186.181	80:30343/TCP,443:30060/TCP	
16m		

5. 在DNS伺服器中新增一個項目、將Astra Control Center CRD檔案中提供的FQDN指向traefik服務的「exter-

IP」。

New Host

Name (uses parent domain name if blank):

astra-control-center

Fully qualified domain name (FQDN):

astra-control-center.cie.netapp.com.

IP address:

10.61.186.181

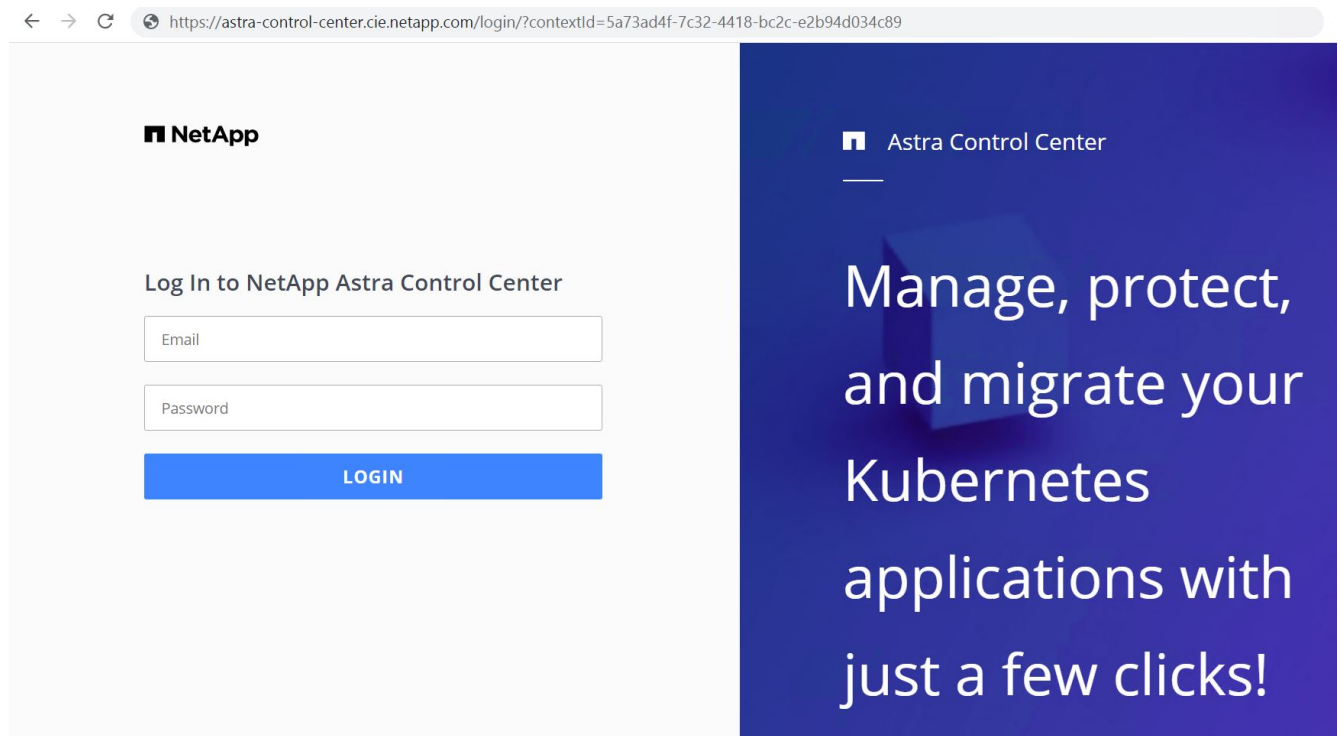
☒ Create associated pointer (PTR) record

☐ Allow any authenticated user to update DNS records with the same owner name

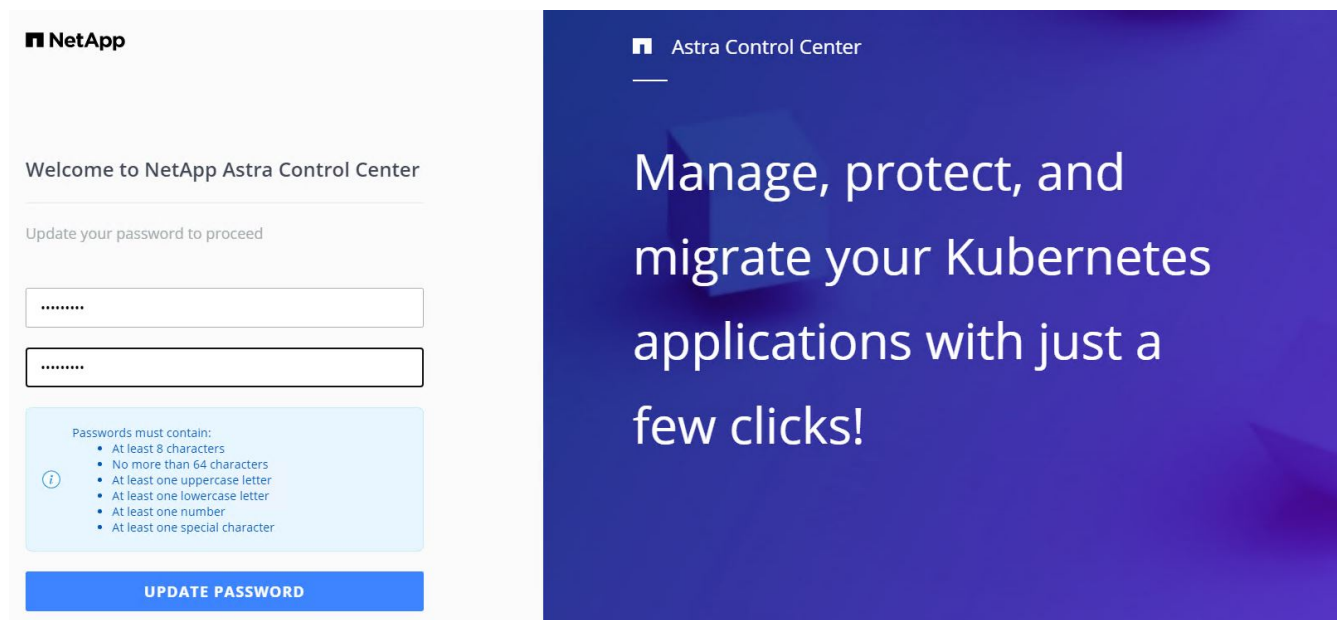
Add Host

Cancel

6. 瀏覽Astra Control Center GUI的FQDN即可登入。

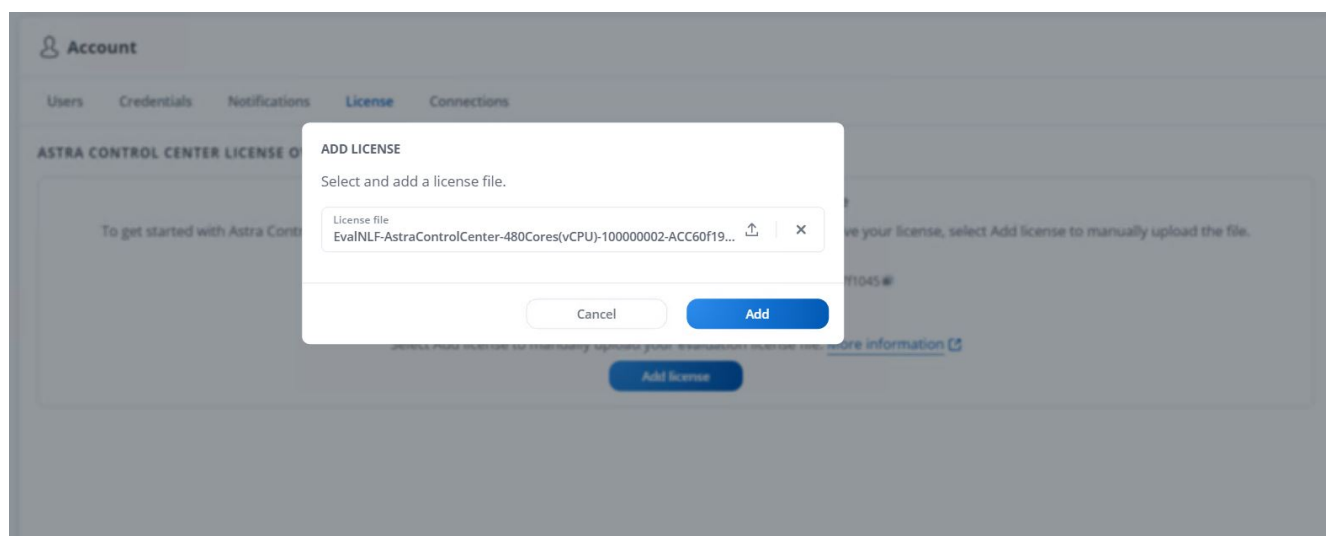


7. 第一次使用CRD提供的管理電子郵件地址登入Astra Control Center GUI時、您需要變更密碼。



8. 如果您想要新增使用者至Astra Control Center、請瀏覽至「帳戶」>「使用者」、按一下「新增」、輸入使用者的詳細資料、然後按一下「新增」。

9. Astra Control Center需要取得授權、才能讓所有IT功能正常運作。若要新增授權、請瀏覽至「帳戶」>「授權」、按一下「新增授權」、然後上傳授權檔案。



如果您在安裝或組態NetApp Astra Control Center時遇到問題、我們將提供已知問題的知識庫 ["請按這裡"](#)。


使用Astra Control Center註冊Red Hat OpenShift叢集

若要讓Astra Control Center管理工作負載、您必須先登錄Red Hat OpenShift叢集。


登錄Red Hat OpenShift叢集

1. 第一步是將OpenShift叢集新增至Astra控制中心、並加以管理。移至「叢集」、然後按一下「新增叢集」、

上傳OpenShift叢集的Kubeconfig檔案、然後按一下「選取儲存設備」。

 **Add cluster**

STEP 1/3: CREDENTIALS



CREDENTIALS



Provide Astra Control access to your Kubernetes and OpenShift clusters by entering a kubeconfig credential.

Follow [instructions](#) on how to create a dedicated admin-role kubeconfig.


[Upload file](#)

Paste from clipboard

Kubeconfig YAML file
ocp-vmw kubeconfig.txt


 

Credential name
ocp-vmw

 **ADDING A CLUSTER**

Adding a cluster is needed for Astra Control to discover your Kubernetes applications.

Select a cloud provider and input credentials to get started.

Read more in [Clusters](#) .

Cancel

Configure storage →



可以產生Kubeconfig檔案、以使用者名稱和密碼或權杖進行驗證。權杖會在有限時間後過期、並可能使登錄的叢集無法連線。NetApp建議使用用戶名和密碼的Kubeconfig檔案、將OpenShift叢集登錄至Astra Control Center。

- Astra Control Center會偵測合格的儲存類別。現在、請在NetApp ONTAP 上選擇使用Trident（以SVM為後盾）來配置Volume的方式、然後按一下「Review（檢閱）」。在下一個窗格中、確認詳細資料、然後按一下「Add Cluster（新增叢集）」。

Add cluster

STEP 2/3: STORAGE

×

STORAGE

Existing storage classes are discovered and verified as eligible for use with Astra Control. You can use your existing default, or choose to set a new default at this time.

Applications with persistent volumes on eligible storage classes are validated for use with Astra Control.

Set default	Storage class	Storage provisioner	Reclaim policy	Binding mode	Eligible
<input checked="" type="radio"/>	ocp-trident Default	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	ocp-trident-iscsi	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	project-1-sc	csi.trident.netapp.io	Delete	Immediate	
<input type="radio"/>	thin	kubernetes.io/vsphere-volume	Delete	Immediate	

← Select credentials

Review →

- 如步驟1所述、登錄兩個OpenShift叢集。新增時、叢集會移至「Discovering」（探索）狀態、而Astra Control Center則會檢查並安裝必要的代理程式。叢集狀態會在成功登錄之後變更為執行中。

admin

10

Dashboard

MANAGE YOUR APPS

Apps

Clusters

MANAGE YOUR STORAGE

Backends

Buckets

MANAGE YOUR ACCOUNT

Account

Activity

Support

Clusters

Actions

+ Add

Search

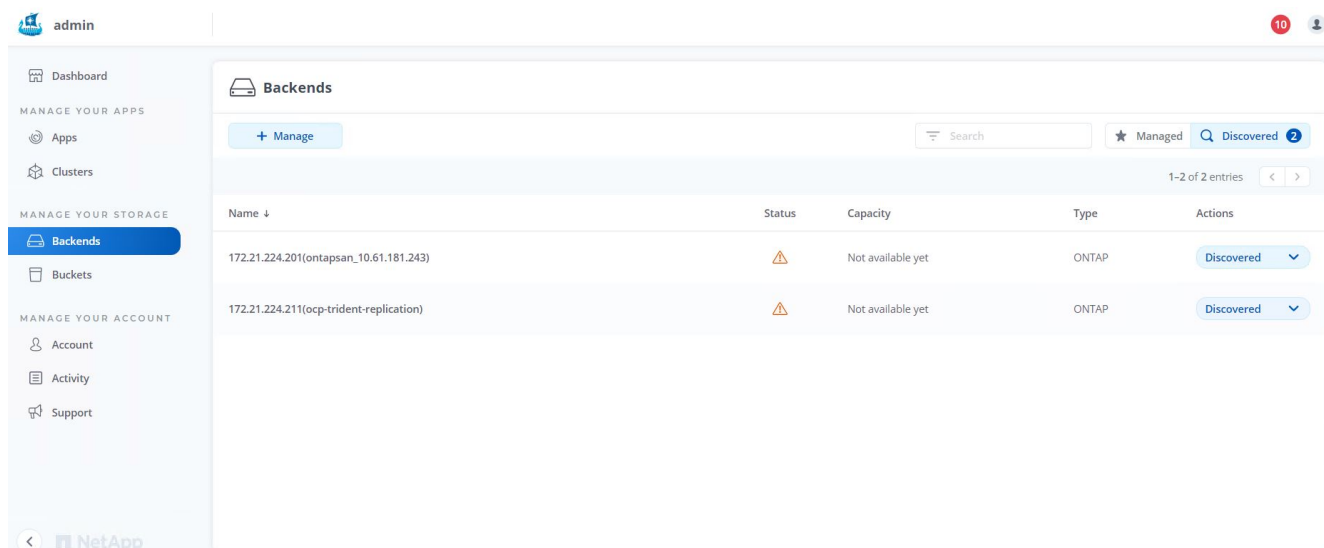
1-2 of 2 entries

<input type="checkbox"/>	Name	Ready	Type	Version	Actions
<input type="checkbox"/>	ocp-vmw		Red Hat OpenShift	v1.20.0+df9c838	Running
<input type="checkbox"/>	ocp-vmware2		Red Hat OpenShift	v1.20.0+c8905da	Running

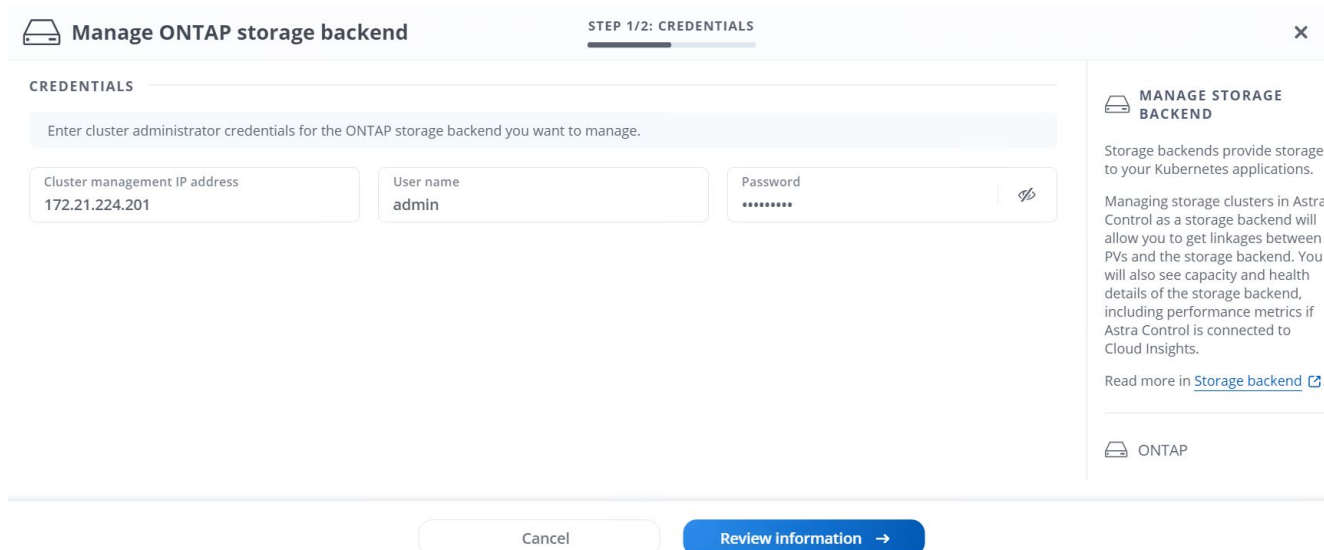


所有由Astra Control Center管理的Red Hat OpenShift叢集、都應該能夠存取安裝時所用的映像登錄、因為安裝在受管理叢集上的代理程式會從該登錄擷取映像。

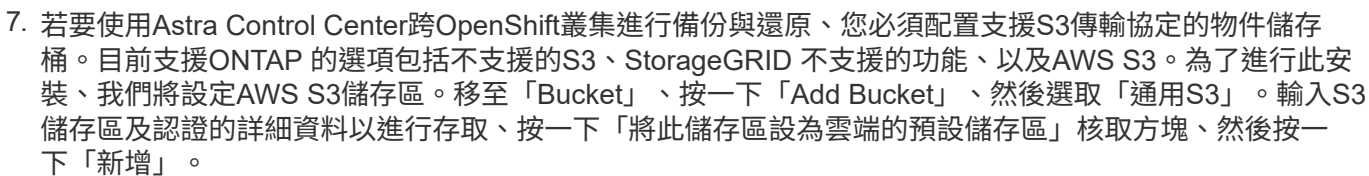
- 將ONTAP 支援的物件叢集匯入為儲存資源、以由Astra Control Center作為後端管理。當OpenShift叢集新增至Astra且已設定儲存設備時、它會自動探索ONTAP 並檢查以儲存設備為後盾的不支援該叢集、但不會將其匯入要管理的Astra Control Center。



- 若要匯入ONTAP 物件叢集、請前往後端、按一下下拉式清單、然後選取ONTAP 要管理之物件叢集旁的「Manage（管理）」。輸入ONTAP 「物件叢集認證」、按一下「檢閱資訊」、然後按一下「匯入儲存設備後端」。



- 新增後端之後、狀態會變更為「可用」。這些後端現在有OpenShift叢集中持續磁碟區的相關資訊、ONTAP 以及在整個系統上對應的磁碟區。

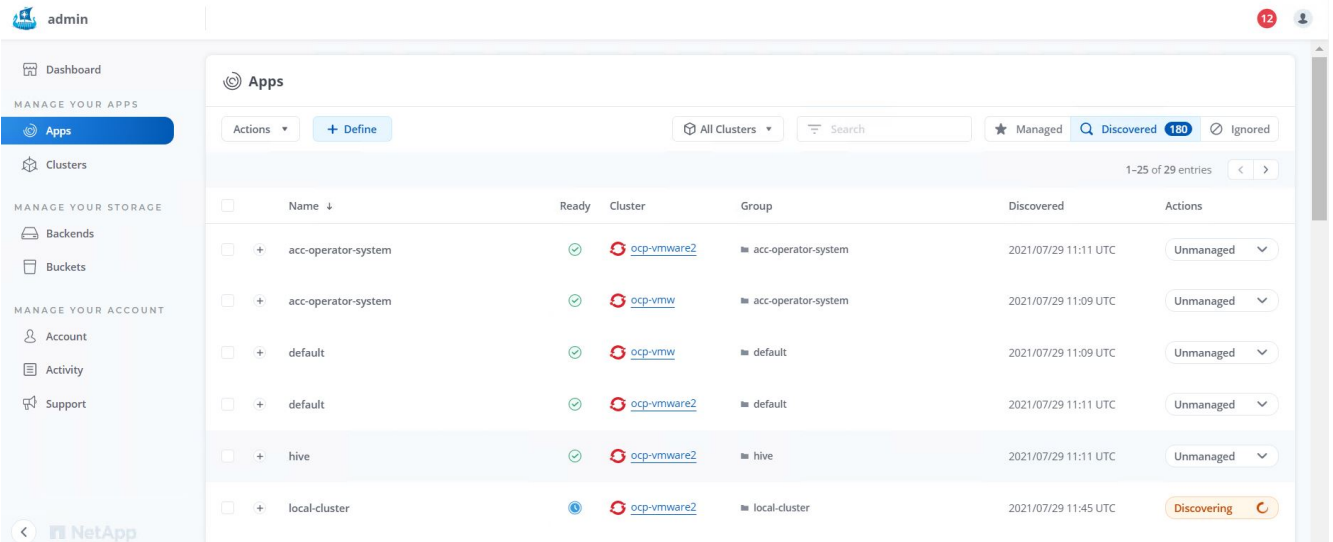


選擇要保護的應用程式

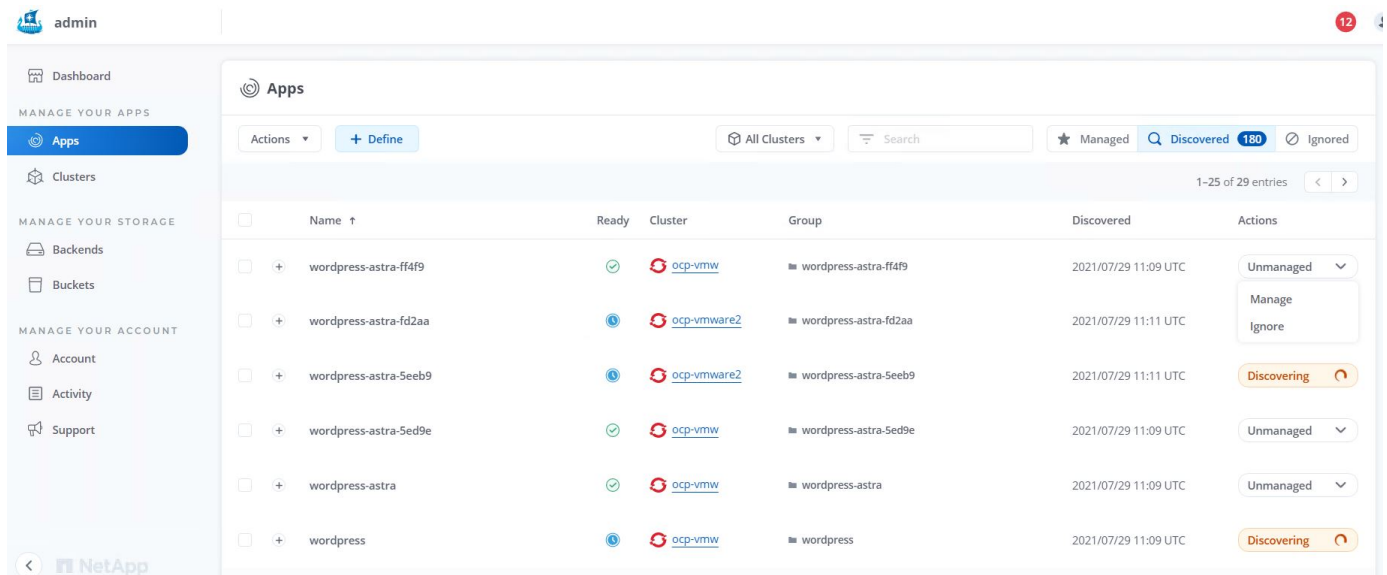
註冊Red Hat OpenShift叢集之後、您可以探索已部署的應用程式、並透過Astra Control Center進行管理。

管理應用程式

1. 在ONTAP Astra Control Center登錄OpenShift叢集和還原後端之後、控制中心會自動開始探索所有命名空間中的應用程式、這些命名空間使用以指定ONTAP 的支援功能後端設定的儲存機架。



2. 瀏覽至「應用程式>已探索」、然後按一下您要使用Astra管理的應用程式旁的下拉式功能表。然後按一下「管理」。



1. 應用程式會進入可用狀態、並可在「應用程式」區段的「受管理」索引標籤下檢視。

Apps

Actions

+ Define

All Clusters

Search

★ Managed

Discovered

175

Ignored

1-1 of 1 entries

<div><input type="checkbox"/></div>	Name	Ready	Protected	Cluster	Group	Discovered	Actions
<div><input type="checkbox"/></div>	wordpress-astra-ff4f9	<div><div></div></div>	<div><div></div></div>	<div><div></div>ocp-vmw</div>	<div><div></div>wordpress-astra-ff4f9</div>	2021/07/29 11:09 UTC	<div>Available</div>

保護您的應用程式

由Astra Control Center管理應用程式工作負載之後、您可以設定這些工作負載的保護設定。

建立應用程式快照

應用程式的快照會建立ONTAP 一份「不含任何資料的Snapshot」複本、以便根據該Snapshot複本、將應用程式還原或複製到特定時間點。

- 若要擷取應用程式的快照、請瀏覽至「應用程式」>「受管理的」索引標籤、然後按一下您要製作Snapshot複本的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下「Snapshot（快照）」。

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images

docker.io/bitnami/mariadb:10.5.13-debian-10-r58
 docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

■ wp

Cluster

Running ▾

Snapshot

Backup

Clone

Restore

Unmanage

- 輸入快照詳細資料、按一下「下一步」、然後按一下「Snapshot（快照）」。建立快照大約需要一分鐘、快照成功建立之後、狀態就會變成可用。

25

Snapshot application

STEP 1/2: DETAILS

✕

SNAPSHOT DETAILS

Name
wp-snapshot-20220228185949

CREATING APPLICATION SNAPSHOTS

Astra Control can take a quick snapshot of your application configuration and persistent storage. Enter a snapshot name to get started.

Read more in [Protect apps](#).

Application

wp

Namespace

wp

Cluster

ocp-vmw

Cancel

Next →

建立應用程式備份

應用程式的備份會擷取應用程式的作用中狀態及其資源組態、將其封裝到檔案中、並將其儲存在遠端物件儲存庫中。

若要在Astra Control Center中備份及還原託管應用程式、您必須先設定支援ONTAP 的支援功能系統的超級使用者設定。若要這麼做、請輸入下列命令。

```
ONTAP::> export-policy rule modify -vserver ocp-trident -policyname default -ruleindex 1 -superuser sys
ONTAP::> export-policy rule modify -policyname default -ruleindex 1 -anon 65534 -vserver ocp-trident
```

1. 若要在Astra Control Center中建立受管理應用程式的備份、請瀏覽至「應用程式」>「受管理的」索引標籤、然後按一下您要備份的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下備份。

wp

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Unprotected

Images
docker.io/bitnami/mariadb:10.5.13-debian-10-r58
docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule
Disabled

Group
wp

Cluster
ocp-vmw

Running

Snapshot
Backup
Clone
Restore
Unmanage

2. 輸入備份詳細資料、選取要保留備份檔案的物件儲存區、按一下「Next（下一步）」、然後在檢閱詳細資料之後、按一下「Backup（備份）」。視應用程式和資料的大小而定、備份可能需要數分鐘的時間、備份成功完成後、備份狀態就會變成可用狀態。

26

Backup application

STEP 1/2: DETAILS

BACKUP DETAILS

Name
wp-backup

☐ Backup from an existing snapshot ?

BACKUP DESTINATION

Bucket
na-ocp-astra/na-ocp-acc Available

CREATING APPLICATION BACKUPS

Astra Control can take a backup of your application configuration and persistent storage. Persistent storage backups are transferred to your object store. Enter a backup name to get started.

Read more in [Application backups](#).

Application
wp

Namespace
wp

Cluster
ocp-vmw

Cancel

Next →

還原應用程式

只要按一下按鈕、就能將應用程式還原至同一個叢集中的原始命名空間、或還原至遠端叢集、以供應用程式保護和災難恢復之用。

- 若要還原應用程式、請瀏覽至「應用程式」>「受管理的」索引標籤、然後按一下有問題的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下「還原」。

wp

APPLICATION STATUS

Healthy

APPLICATION PROTECTION STATUS

Partially protected

Images

- docker.io/bitnami/mariadb:10.5.13-debian-10-r58
- docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule

Disabled

Group

wp

Cluster

Running

- Snapshot
- Backup
- Clone
- Restore
- Unmanage

- 輸入還原命名空間的名稱、選取您要還原的叢集、然後選擇是要從現有的快照或應用程式備份還原命名空間。按一下「下一步」

Restore application

STEP 1/2: DETAILS

RESTORE DETAILS

Destination cluster

ocp-vmw

Destination namespace

wp

RESTORE SOURCE

Filter

Snapshots Backups

Application backup	Ready	On-Schedule/On-Demand	Created ↑
wp-backup	✓	On-Demand	2022/02/28 18:54 UTC

RESTORING APPLICATIONS

Astra Control can restore your application configuration and persistent storage. Select a source snapshot or backup for the restored application.

- Application wp
- Namespace wp
- Cluster ocp-vmw

Cancel

Next →

- 在檢閱窗格中、輸入「重新儲存」、並在檢閱詳細資料後按一下「還原」。

Restore application

STEP 2/2: SUMMARY

REVIEW RESTORE INFORMATION

⚠

All existing resources associated with this application will be deleted and replaced with the source backup "wp-backup" taken on 2022/02/28 18:54 UTC. Persistent volumes will be deleted and recreated. External resources with dependencies on this application may be impacted.

We recommend taking a snapshot or a backup of your application before proceeding.

BACKUP

wp-backup

ORIGINAL GROUP

wp

ORIGINAL CLUSTER

ocp-vmw

RESOURCE LABELS

ClusterRole
kubernetes.io/bootstrapping: rbac-defaults +1
ClusterRoleBinding

RESTORE

wp

DESTINATION GROUP

wp

DESTINATION CLUSTER

ocp-vmw

RESOURCE LABELS

ClusterRole
kubernetes.io/bootstrapping: rbac-defaults +1
ClusterRoleBinding

Are you sure you want to restore the application "wp"?

Type **restore** below to confirm.

Confirm to restore

restore

← Back

Restore ✓

- 新應用程式會在Astra Control Center還原所選叢集上的應用程式時、進入還原狀態。Astra安裝並偵測應用程式的所有資源之後、應用程式會進入可用狀態。

Actions ▾

+ Define

▾

Search

★

🔍

110

🗑

🔄

1-1 of 1 entries

<


>



<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wp	✔	<div><div></div><div>i</div></div>	<div><div></div><div>ocp-vmw</div></div>	<div><div></div><div>wp</div></div>	2022/02/28 18:34 UTC	<div>Available</div> <div>▾</div>



複製應用程式

您可以將應用程式複製到原始叢集或遠端叢集、以供開發/測試或應用程式保護及災難恢復之用。在同一個儲存後端的同一個叢集內複製應用程式時、會使用NetApp FlexClone技術來即時複製PVCS、並節省儲存空間。

- 若要複製應用程式、請瀏覽至「應用程式」>「受管理」索引標籤、然後按一下有問題的應用程式。按一下應用程式名稱旁的下拉式功能表、然後按一下Clone（複製）。


 wp


 APPLICATION STATUS
 Healthy


 APPLICATION PROTECTION STATUS
 Partially protected

Images
 docker.io/bitnami/mariadb:10.5.13-debian-10-r58
 docker.io/bitnami/wordpress:5.9.0-debian-10-r1

Protection schedule
 Disabled


Group
 wp

 Cluster
 ocp-vmw

Running 

Snapshot
 Backup
 Clone
 Restore
 Unmanage

- 輸入新命名空間的詳細資料、選取要複製到的叢集、然後選擇是否要從現有的快照或備份複製、或是從應用程式的目前狀態複製。檢閱詳細資料後、按一下「下一步」、然後按一下「檢閱窗格上的Clone（複製）」。

 Clone application


STEP 1/2: DETAILS


✕

CLONE DETAILS

Clone name
 wp-clone

Clone namespace
 wp-clone


Destination cluster
 ocp-vmw


☐ Clone from an existing snapshot or backup 


CLONING APPLICATIONS

Astra Control can create a clone of your application configuration and persistent storage. Persistent storage backups are transferred from your object store, so choosing a clone from an existing backup will complete the fastest. Enter a clone name to get started.

Read more in [Clone applications](#).

 Application
wp

 Namespace
wp

 Cluster
ocp-vmw

Cancel

Next →

- 當Astra Control Center在所選叢集上建立應用程式時、新的應用程式會進入「Discovering（探索）」狀態。Astra安裝並偵測應用程式的所有資源之後、應用程式會進入可用狀態。

Actions ▾

+ Define

📦 ▾

🔍 Search

★

🔍

110




🗑

🔄

1-2 of 2 entries

<

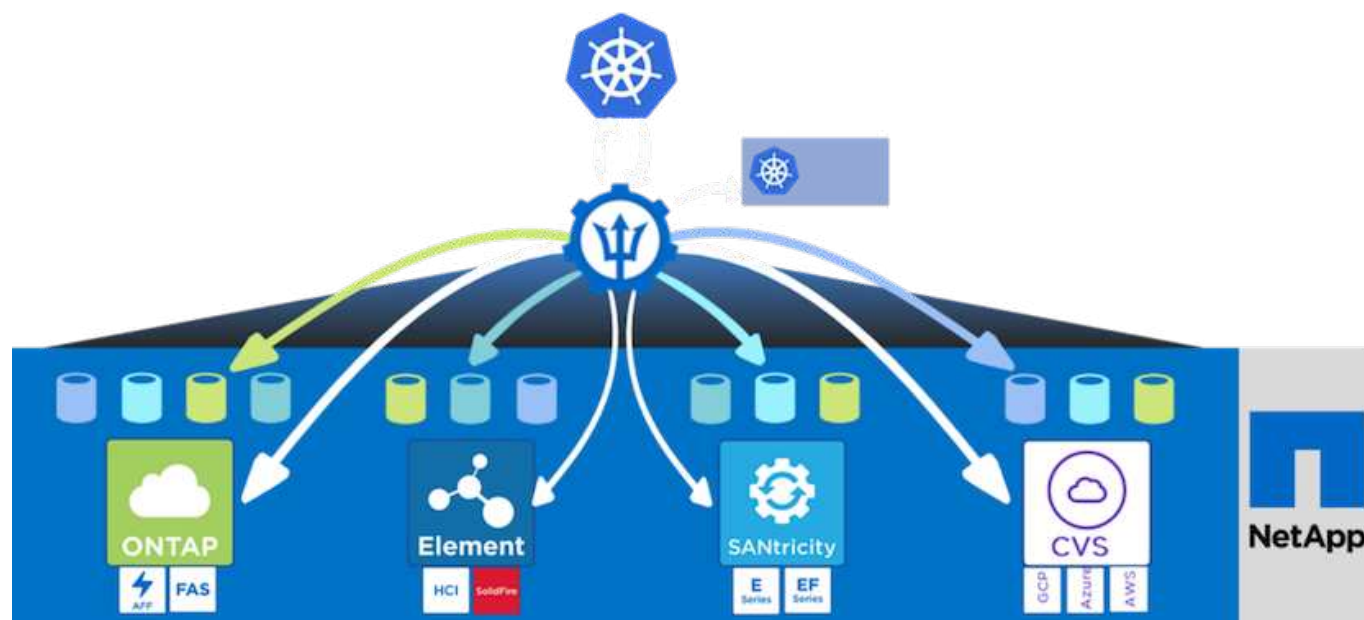
>

<input type="checkbox"/>	Name ↓	Ready	Protected	Cluster	Group	Discovered	Actions
<input type="checkbox"/>	wp	✓	i	 ocp-vmw	📁 wp	2022/02/28 18:34 UTC	Available ▾
<input type="checkbox"/>	wp-clone	✓		 ocp-vmw	📁 wp-clone	2022/02/28 19:21 UTC	Available ▾

Astra Trident總覽

Astra Trident是開放原始碼且完全支援的儲存協調工具、適用於Container和Kubernetes配送、包括Red Hat OpenShift。Trident可搭配整個NetApp儲存產品組合（包括NetApp ONTAP 的整套和Element儲存系統）使用、也支援NFS和iSCSI連線。Trident可讓終端使用者從NetApp儲存系統配置及管理儲存設備、而無需儲存管理員介入、進而加速DevOps工作流程。

系統管理員可根據專案需求和儲存系統模型來設定多個儲存後端、以啟用進階儲存功能、包括壓縮、特定磁碟類型或QoS層級、以保證特定層級的效能。定義後端後端之後、開發人員可在專案中使用這些後端來建立持續磁碟區宣告（PVCS）、並視需要將持續儲存附加至容器。



Astra Trident的開發週期很快、就像Kubernetes一樣、每年發行四次。

最新版的Astra Trident於2022年1月推出22.01版。支援對照表、顯示哪些版本的Trident已通過測試、可找到Kubernetes經銷產品 ["請按這裡"](#)。

從20.04版本開始、Trident設定由Trident操作員執行。營運者讓大規模部署變得更簡單、並提供額外支援、包括在Trident安裝過程中部署的Pod自我修復。

有了21.01版、我們提供了Helm圖表、方便您安裝Trident操作員。

下載Astra Trident

若要在已部署的使用者叢集上安裝Trident並佈建持續磁碟區、請完成下列步驟：

1. 將安裝歸檔檔案下載至管理工作站、並擷取內容。目前的Trident版本為22.01、可下載 ["請按這裡"](#)。

```
[netapp-user@rhel7 ~]$ wget
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
--2021-05-06 15:17:30--
https://github.com/NetApp/trident/releases/download/v22.01.0/trident-
installer-22.01.0.tar.gz
Resolving github.com (github.com)... 140.82.114.3
Connecting to github.com (github.com)|140.82.114.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
22.01.0.tar.gz&response-content-type=application%2Foctet-stream
[following]
--2021-05-06 15:17:30-- https://github-
releases.githubusercontent.com/77179634/a4fa9f00-a9f2-11eb-9053-
98e8e573d4ae?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAIWNJYAX4CSVEH53A%2F20210506%2Fus-east-
1%2Fs3%2Faws4_request&X-Amz-Date=20210506T191643Z&X-Amz-Expires=300&X-
Amz-
Signature=8a49a2a1e08c147d1ddd8149ce45a5714f9853fee19bb1c507989b9543eb36
30&X-Amz-
SignedHeaders=host&actor_id=0&key_id=0&repo_id=77179634&response-
content-disposition=attachment%3B%20filename%3Dtrident-installer-
22.01.0.tar.gz&response-content-type=application%2Foctet-stream
Resolving github-releases.githubusercontent.com (github-
releases.githubusercontent.com)... 185.199.108.154, 185.199.109.154,
185.199.110.154, ...
Connecting to github-releases.githubusercontent.com (github-
releases.githubusercontent.com)|185.199.108.154|:443... connected.
HTTP request sent, awaiting response... 200 OK
```

```
Length: 38349341 (37M) [application/octet-stream]
Saving to: 'trident-installer-22.01.0.tar.gz'

100%[=====
=====>] 38,349,341  88.5MB/s
in 0.4s

2021-05-06 15:17:30 (88.5 MB/s) - 'trident-installer-22.01.0.tar.gz'
saved [38349341/38349341]
```

2. 從下載的套裝組合中擷取Trident安裝。

```
[netapp-user@rhel7 ~]$ tar -xzf trident-installer-22.01.0.tar.gz
[netapp-user@rhel7 ~]$ cd trident-installer/
[netapp-user@rhel7 trident-installer]$
```

使用Helm安裝Trident運算子

1. 首先將使用者叢集的「kubeconfig」檔案位置設定為環境變數、這樣您就不需要參考它、因為Trident沒有傳遞此檔案的選項。

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/.ocp-
install/auth/kubeconfig
```

2. 在使用者叢集中建立Trident命名空間時、請執行Helm命令、從Lm目錄的tar安裝Trident運算子。

```
[netapp-user@rhel7 trident-installer]$ helm install trident
helm/trident-operator-22.01.0.tgz --create-namespace --namespace trident
NAME: trident
LAST DEPLOYED: Fri May  7 12:54:25 2021
NAMESPACE: trident
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
Thank you for installing trident-operator, which will deploy and manage
NetApp's Trident CSI
storage provisioner for Kubernetes.

Your release is named 'trident' and is installed into the 'trident'
namespace.
Please note that there must be only one instance of Trident (and
trident-operator) in a Kubernetes cluster.

To configure Trident to manage storage resources, you will need a copy
of tridentctl, which is
available in pre-packaged Trident releases. You may find all Trident
releases and source code
online at https://github.com/NetApp/trident.

To learn more about the release, try:

$ helm status trident
$ helm get all trident
```

3. 您可以檢查在命名空間中執行的Pod、或使用tridentctl二進位檔檢查安裝的版本、以驗證Trident是否已成功安裝。

```
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
```

NAME	READY	STATUS	RESTARTS	AGE
trident-csi-5z451	1/2	Running	2	30s
trident-csi-696b685cf8-htdb2	6/6	Running	0	30s
trident-csi-b74p2	2/2	Running	0	30s
trident-csi-lrw4n	2/2	Running	0	30s
trident-operator-7c748d957-gr2gw	1/1	Running	0	36s

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
```

```
+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+
| 22.01.0        | 22.01.0        |
+-----+
```



在某些情況下、客戶環境可能需要自訂Trident部署。在這些情況下、您也可以手動安裝Trident運算子、並更新隨附的資訊清單、以自訂部署。

手動安裝Trident運算子

1. 首先、將使用者叢集的「kubeconfig」檔案位置設定為環境變數、這樣您就不需要參考、因為Trident沒有傳遞此檔案的選項。

```
[netapp-user@rhel7 trident-installer]$ export KUBECONFIG=~/.ocp-
install/auth/kubeconfig
```

2. 「Trident安裝程式」目錄包含定義所有必要資源的資訊清單。使用適當的資訊清單、建立「TridentOrchestrator」自訂資源定義。

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/trident.netapp.io_tridentorchestrators_crd_post1.16.yaml
customresourcedefinition.apiextensions.k8s.io/tridentorchestrators.tride
nt.netapp.io created
```

3. 如果不存在、請使用提供的資訊清單、在叢集中建立Trident命名空間。

```
[netapp-user@rhel7 trident-installer]$ oc apply -f deploy/namespace.yaml
namespace/trident created
```

4. 建立Trident營運者部署所需的資源、例如營運者的「服務帳戶」、專屬的「PodSecurity Policy」、或營運者本身的「ClusterRole」和「ClusterRoleBinding」。

```
[netapp-user@rhel7 trident-installer]$ oc create -f deploy/bundle.yaml
serviceaccount/trident-operator created
clusterrole.rbac.authorization.k8s.io/trident-operator created
clusterrolebinding.rbac.authorization.k8s.io/trident-operator created
deployment.apps/trident-operator created
podsecuritypolicy.policy/tridentoperatorpods created
```

5. 您可以使用下列命令來檢查部署後的操作員狀態：

```
[netapp-user@rhel7 trident-installer]$ oc get deployment -n trident
NAME                READY    UP-TO-DATE    AVAILABLE    AGE
trident-operator    1/1      1              1            23s
[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY    STATUS    RESTARTS    AGE
trident-operator-66f48895cc-lzczk    1/1      Running    0           41s
```

6. 部署營運者之後、我們就可以使用它來安裝Trident。這需要建立「TridentOrchestrator」。

```
[netapp-user@rhel7 trident-installer]$ oc create -f
deploy/crds/tridentorchestrator_cr.yaml
tridentorchestrator.trident.netapp.io/trident created
[netapp-user@rhel7 trident-installer]$ oc describe torc trident
Name:                trident
Namespace:
Labels:               <none>
Annotations:          <none>
API Version:          trident.netapp.io/v1
Kind:                 TridentOrchestrator
Metadata:
  Creation Timestamp:  2021-05-07T17:00:28Z
  Generation:          1
  Managed Fields:
    API Version:        trident.netapp.io/v1
    Fields Type:        FieldsV1
    fieldsV1:
      f:spec:
        .:
        f:debug:
        f:namespace:
  Manager:             kubect1-create
  Operation:            Update
  Time:                 2021-05-07T17:00:28Z
  API Version:          trident.netapp.io/v1
```

```

Fields Type:  FieldsV1
fieldsV1:
  f:status:
    .:
  f:currentInstallationParams:
    .:
    f:IPv6:
    f:autosupportHostname:
    f:autosupportImage:
    f:autosupportProxy:
    f:autosupportSerialNumber:
    f:debug:
    f:enableNodePrep:
    f:imagePullSecrets:
    f:imageRegistry:
    f:k8sTimeout:
    f:kubeletDir:
    f:logFormat:
    f:silenceAutosupport:
    f:tridentImage:
  f:message:
  f:namespace:
  f:status:
  f:version:
Manager:      trident-operator
Operation:    Update
Time:         2021-05-07T17:00:28Z
Resource Version: 931421
Self Link:
/apis/trident.netapp.io/v1/tridentorchestrators/trident
UID:          8a26a7a6-dde8-4d55-9b66-a7126754d81f
Spec:
  Debug:      true
  Namespace:  trident
Status:
  Current Installation Params:
    IPv6:          false
    Autosupport Hostname:
    Autosupport Image:      netapp/trident-autosupport:21.01
    Autosupport Proxy:
    Autosupport Serial Number:
    Debug:          true
    Enable Node Prep:      false
    Image Pull Secrets:
    Image Registry:
    k8sTimeout:      30

```

```

Kubelet Dir:      /var/lib/kubelet
Log Format:       text
Silence Autosupport: false
Trident Image:    netapp/trident:22.01.0
Message:         Trident installed
Namespace:       trident
Status:          Installed
Version:         v22.01.0
Events:
  Type    Reason      Age   From                                Message
  ----    -
Normal    Installing  80s   trident-operator.netapp.io         Installing
Trident
Normal    Installed  68s   trident-operator.netapp.io         Trident
installed

```

7. 您可以檢查在命名空間中執行的Pod、或使用tridentctl二進位檔檢查安裝的版本、以驗證Trident是否已成功安裝。

```

[netapp-user@rhel7 trident-installer]$ oc get pods -n trident
NAME                                READY   STATUS    RESTARTS   AGE
trident-csi-bb64c6cb4-lmd6h        6/6     Running   0           82s
trident-csi-gn59q                  2/2     Running   0           82s
trident-csi-m4szj                  2/2     Running   0           82s
trident-csi-sb9k9                  2/2     Running   0           82s
trident-operator-66f48895cc-lzczk  1/1     Running   0           2m39s

[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident version
+-----+-----+
| SERVER VERSION | CLIENT VERSION |
+-----+-----+
| 22.01.0        | 22.01.0        |
+-----+-----+

```

準備工作節點以供儲存

NFS

大多數Kubernetes發佈版本都隨附套件和公用程式、可在預設情況下安裝NFS後端、包括Red Hat OpenShift。

不過、對於NFSv3、用戶端與伺服器之間沒有協調並行的機制。因此、用戶端SUNRPC插槽表項目的最大數量必須以伺服器上支援的值手動同步、以確保NFS連線的最佳效能、而無需伺服器減少連線的視窗大小。

對於支援的SUNRPC插槽表項目數量上限為128、亦即、支援的每次可同時處理128個NFS要求。ONTAP
ONTAP不過、根據預設、每個連線的Red Hat CoreOS/Red Hat Enterprise Linux最多可有65536個SUNRPC插槽表項目。我們需要將此值設為128、這可透過OpenShift中的機器組態操作員（MCO）來完成。

若要修改OpenShift工作節點中的最大社工PC插槽表格項目、請完成下列步驟：

1. 登入OCP網路主控台、然後瀏覽至「運算」>「機器組態」。按一下「Create Machine Config（建立機器組複製並貼上Yaml檔案、然後按一下「Create（建立）」。

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 98-worker-nfs-rpc-slot-tables
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - contents:
            source: data:text/plain;charset=utf-8;base64,b3B0aW9ucyBzdW5ycGMgdGNwX21heF9zbG90X3RhYmxlX2VudHJpZXM9MTI4Cg==
            filesystem: root
            mode: 420
            path: /etc/modprobe.d/sunrpc.conf
```

2. 建立MCO之後、必須在所有工作節點上套用組態、然後逐一重新開機。整個程序約需20至30分鐘。使用「oce Get MCP」確認是否套用機器組態、並確認已更新員工的機器組態集區。

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
```

NAME	CONFIG	UPDATED	UPDATING
DEGRADED			
master	rendered-master-a520ae930e1d135e0dee7168	True	False
False			
worker	rendered-worker-de321b36eeba62df41feb7bc	True	False
False			

iSCSI

若要準備工作節點、以便透過iSCSI傳輸協定對應區塊儲存磁碟區、您必須安裝必要的套件、才能支援該功能。

在Red Hat OpenShift中、這是在叢集部署之後、將MCO（機器組態操作員）套用至叢集來處理。

若要設定工作節點以執行iSCSI服務、請完成下列步驟：

1. 登入OCP網路主控台、然後瀏覽至「運算」>「機器組態」。按一下「Create Machine Config（建立機器組

複製並貼上Yaml檔案、然後按一下「Create（建立）」。

不使用多重路徑時：

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 99-worker-element-iscsi
spec:
  config:
    ignition:
      version: 3.2.0
    systemd:
      units:
        - name: iscsid.service
          enabled: true
          state: started
  osImageURL: ""
```

使用多重路徑時：

```

apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 99-worker-ontap-iscsi
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
      - contents:
          source: data:text/plain;charset=utf-8;base64,ZGVmYXVsdHMgewogICAgICAgIHVzZXJfZnJpZW5kbHlfbmFtZXMgYm8KICAgICAgICBmaW5kX211bHRpcGF0aHMgYm8KfQoKYmxhY2tsaXN0X2V4Y2VwdGlvbnMgewogICAgICAgIHByb3BlcnR5ICIoU0NTSV9JREVOVF98SURfV1dOKSfQoKYmxhY2tsaXN0IHsKfQoK
          verification: {}
        filesystem: root
        mode: 400
        path: /etc/multipath.conf
    systemd:
      units:
      - name: iscsid.service
        enabled: true
        state: started
      - name: multipathd.service
        enabled: true
        state: started
  osImageURL: ""

```

2. 建立組態之後、將組態套用至工作節點並重新載入大約需要20到30分鐘的時間。使用「ocean Get MCP」確認是否套用機器組態、並確認已更新員工的機器組態集區。您也可以登入工作者節點、確認iscsid服務正在執行（如果使用多重路徑、則多路徑服務正在執行）。

```
[netapp-user@rhel7 openshift-deploy]$ oc get mcp
NAME          CONFIG                                UPDATED    UPDATING
DEGRADED
master        rendered-master-a520ae930e1d135e0dee7168    True       False
False
worker        rendered-worker-de321b36eeba62df41feb7bc    True       False
False

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status iscsid
● iscsid.service - Open-iSCSI
   Loaded: loaded (/usr/lib/systemd/system/iscsid.service; enabled;
   vendor preset: disabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
     Docs: man:iscsid(8)
           man:iscsiadm(8)
  Main PID: 1242 (iscsid)
    Status: "Ready to process requests"
     Tasks: 1
  Memory: 4.9M
     CPU: 9ms
   CGroup: /system.slice/iscsid.service
           └─1242 /usr/sbin/iscsid -f

[netapp-user@rhel7 openshift-deploy]$ ssh core@10.61.181.22 sudo
systemctl status multipathd
● multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled;
   vendor preset: enabled)
   Active: active (running) since Tue 2021-05-26 13:36:22 UTC; 3 min ago
  Main PID: 918 (multipathd)
    Status: "up"
     Tasks: 7
  Memory: 13.7M
     CPU: 57ms
   CGroup: /system.slice/multipathd.service
           └─918 /sbin/multipathd -d -s
```



此外、您也可以使用適當的旗標來執行「occ偵錯」命令、確認機器組態已成功套用、服務已如預期般啟動。

建立儲存系統後端

完成Astra Trident操作員安裝之後、您必須為所使用的特定NetApp儲存平台設定後端。請依照下列連結繼續Astra Trident的設定與組態。

- ["NetApp ONTAP 不適用於NFS"](#)
- ["NetApp ONTAP 支援iSCSI"](#)
- ["支援iSCSI NetApp Element"](#)

NetApp ONTAP 不適用於NFS組態

若要與NetApp ONTAP 支援儲存系統進行Trident整合、您必須建立後端、以便與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。對於ONTAP NetApp支援NFS的系統、請將「backender-ontap - nas.json」檔案複製到您的工作目錄、然後編輯檔案。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/ontap-nas/backend-ontap-nas.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-nas.json
```

2. 編輯後端名稱、管理LIF、dataLIF、SVM、使用者名稱、和密碼值。

```
{
  "version": 1,
  "storageDriverName": "ontap-nas",
  "backendName": "ontap-nas+10.61.181.221",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.221",
  "svm": "trident_svm",
  "username": "cluster-admin",
  "password": "password"
}
```



最佳實務做法是將自訂的backendName值定義為storageDriverName和資料LIF的組合、以利NFS識別。

3. 在這個後端檔案就緒的情況下、執行下列命令來建立第一個後端。

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-nas.json
```

NAME		STORAGE DRIVER	UUID
STATE	VOLUMES		
ontap-nas+10.61.181.221	ontap-nas	be7a619d-c81d-445c-b80c-5c87a73c5b1e	online 0

4. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.template ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-nas"
```



在此檔案中定義了一個名為「FSType」的選用欄位。此行可在NFS後端刪除。

6. 執行「oc」命令以建立儲存類別。

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

7. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告 (PVC)。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

8. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVC定義。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

9. 使用「oc」命令建立PVC。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME      STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
basic      Bound       pvc-b4370d37-0fa4-4c17-bd86-94f96c94b42d  1Gi
RWO          basic-csi     7s
```

NetApp ONTAP 支援iSCSI組態

若要與NetApp ONTAP 支援儲存系統進行Trident整合、您必須建立後端、以便與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。對於ONTAP 供應iSCSI的NetApp 支援系統、請將「backender-ontap - san . json」檔案複製到您的工作目錄、然後編輯該檔案。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-samples/ontap-san/backend-ontap-san.json ./
[netapp-user@rhel7 trident-installer]$ vi backend-ontap-san.json
```

2. 編輯此檔案中的管理LIF、dataLIF、SVM、使用者名稱和密碼值。

```
{
  "version": 1,
  "storageDriverName": "ontap-san",
  "managementLIF": "172.21.224.201",
  "dataLIF": "10.61.181.240",
  "svm": "trident_svm",
  "username": "admin",
  "password": "password"
}
```

3. 在這個後端檔案就緒的情況下、執行下列命令來建立第一個後端。

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-ontap-san.json
+-----+-----+
+-----+-----+-----+-----+
|          NAME          | STORAGE DRIVER |                      UUID                      |
| STATE | VOLUMES | |                      |                      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
| ontapsan_10.61.181.241 | ontap-san      | 6788533c-7fea-4a35-b797- |
| fb9bb3322b91 | online | 0 |                      |
+-----+-----+-----+-----+
+-----+-----+-----+-----+
```

4. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.tmpl ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

5. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

```

apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "ontap-san"

```



在此檔案中定義了一個名為「FSType」的選用欄位。在iSCSI後端中、此值可設定為特定的Linux檔案系統類型（XFS、ext4等）、也可刪除以允許OpenShift決定要使用的檔案系統。

6. 執行「oc」命令以建立儲存類別。

```

[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created

```

7. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告（PVC）。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

```

[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-
basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml

```

8. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVC定義。

```

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi

```

9. 使用「oc」命令建立PVC。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。


```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME      STATUS    VOLUME                                     CAPACITY
ACCESS MODES   STORAGECLASS  AGE
basic       Bound        pvc-7ceac1ba-0189-43c7-8f98-094719f7956c  1Gi
RWO          basic-csi     3s
```

支援iSCSI組態NetApp Element

若要啟用Trident與NetApp Element 支援功能的整合、您必須建立後端、以便使用iSCSI傳輸協定與儲存系統進行通訊。

1. 下載的安裝歸檔文件中有「shame-INPUT」資料夾階層的範例後端檔案。若NetApp Element 為供應iSCSI的支援系統、請將「backend-solidfire.json」檔案複製到您的工作目錄、然後編輯檔案。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/backends-
samples/solidfire/backend-solidfire.json ./
[netapp-user@rhel7 trident-installer]$ vi ./backend-solidfire.json
```

- a. 編輯「端點」行上的使用者、密碼和MVIP值。
- b. 編輯「VIP」值。

```
{
  "version": 1,
  "storageDriverName": "solidfire-san",
  "Endpoint": "https://trident:password@172.21.224.150/json-
rpc/8.0",
  "SVIP": "10.61.180.200:3260",
  "TenantName": "trident",
  "Types": [{"Type": "Bronze", "Qos": {"minIOPS": 1000, "maxIOPS":
2000, "burstIOPS": 4000}},
            {"Type": "Silver", "Qos": {"minIOPS": 4000, "maxIOPS":
6000, "burstIOPS": 8000}},
            {"Type": "Gold", "Qos": {"minIOPS": 6000, "maxIOPS":
8000, "burstIOPS": 10000}}]
}
```

2. 在這個後端檔案就位的情況下、執行下列命令來建立您的第一個後端。

```
[netapp-user@rhel7 trident-installer]$ ./tridentctl -n trident create
backend -f backend-solidfire.json
```

NAME	STATE	VOLUMES	STORAGE DRIVER	UUID
solidfire_10.61.180.200	online	0	solidfire-san	b90783ee-e0c9-49af-8d26-3ea87ce2efdf

3. 建立後端之後、您必須接著建立儲存類別。就像後端一樣、範例輸入資料夾中也有可供編輯的儲存類別檔案範例。將其複製到工作目錄、並進行必要的編輯、以反映所建立的後端。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/storage-class-
samples/storage-class-csi.yaml.template ./storage-class-basic.yaml
[netapp-user@rhel7 trident-installer]$ vi storage-class-basic.yaml
```

4. 唯一必須對此檔案進行的編輯、是從新建立的後端、將「backendType」值定義為儲存驅動程式名稱。另請注意名稱欄位值、此值必須在後續步驟中參考。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: basic-csi
provisioner: csi.trident.netapp.io
parameters:
  backendType: "solidfire-san"
```



在此檔案中定義了一個名為「FSType」的選用欄位。在iSCSI後端中、此值可設定為特定的Linux檔案系統類型（XFS、ext4等）、也可刪除此值、讓OpenShift決定要使用的檔案系統。

5. 執行「oc」命令以建立儲存類別。

```
[netapp-user@rhel7 trident-installer]$ oc create -f storage-class-
basic.yaml
storageclass.storage.k8s.io/basic-csi created
```

6. 建立儲存類別之後、您必須建立第一個持續磁碟區宣告（PVC）。還有一個「PVC-base.yaml」檔案範例、也可在範例輸入中執行此動作。

```
[netapp-user@rhel7 trident-installer]$ cp sample-input/pvc-samples/pvc-basic.yaml ./
[netapp-user@rhel7 trident-installer]$ vi pvc-basic.yaml
```

7. 唯一必須對此檔案進行的編輯、是確保「儲存類別名稱」欄位符合剛剛建立的欄位。您可以根據要配置的工作負載需求、進一步自訂PVC定義。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: basic
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 1Gi
  storageClassName: basic-csi
```

8. 使用「oc」命令建立PVC。視所建立的備用磁碟區大小而定、建立作業可能需要一些時間、因此您可以在完成時觀看程序。

```
[netapp-user@rhel7 trident-installer]$ oc create -f pvc-basic.yaml
persistentvolumeclaim/basic created

[netapp-user@rhel7 trident-installer]$ oc get pvc
NAME      STATUS    VOLUME                                     CAPACITY
ACCESS MODES  STORAGECLASS  AGE
basic      Bound       pvc-3445b5cc-df24-453d-a1e6-b484e874349d  1Gi
RWO          basic-csi     5s
```

版權資訊

Copyright © 2024 NetApp, Inc. 版權所有。台灣印製。非經版權所有人事先書面同意，不得將本受版權保護文件的任何部分以任何形式或任何方法（圖形、電子或機械）重製，包括影印、錄影、錄音或儲存至電子檢索系統中。

由 NetApp 版權資料衍伸之軟體必須遵守下列授權和免責聲明：

此軟體以 NETAPP「原樣」提供，不含任何明示或暗示的擔保，包括但不限於有關適售性或特定目的適用性之擔保，特此聲明。於任何情況下，就任何已造成或基於任何理論上責任之直接性、間接性、附隨性、特殊性、懲罰性或衍生性損害（包括但不限於替代商品或服務之採購；使用、資料或利潤上的損失；或企業營運中斷），無論是在使用此軟體時以任何方式所產生的契約、嚴格責任或侵權行為（包括疏忽或其他）等方面，NetApp 概不負責，即使已被告知有前述損害存在之可能性亦然。

NetApp 保留隨時變本文所述之任何產品的權利，恕不另行通知。NetApp 不承擔因使用本文所述之產品而產生的責任或義務，除非明確經過 NetApp 書面同意。使用或購買此產品並不會在依據任何專利權、商標權或任何其他 NetApp 智慧財產權的情況下轉讓授權。

本手冊所述之產品受到一項（含）以上的美國專利、國外專利或申請中專利所保障。

有限權利說明：政府機關的使用、複製或公開揭露須受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中的「技術資料權利 - 非商業項目」條款 (b)(3) 小段所述之限制。

此處所含屬於商業產品和 / 或商業服務（如 FAR 2.101 所定義）的資料均為 NetApp, Inc. 所有。根據本協議提供的所有 NetApp 技術資料和電腦軟體皆屬於商業性質，並且完全由私人出資開發。美國政府對於該資料具有非專屬、非轉讓、非轉授權、全球性、有限且不可撤銷的使用權限，僅限於美國政府為傳輸此資料所訂合約所允許之範圍，並基於履行該合約之目的方可使用。除非本文另有規定，否則未經 NetApp Inc. 事前書面許可，不得逕行使用、揭露、重製、修改、履行或展示該資料。美國政府授予國防部之許可權利，僅適用於 DFARS 條款 252.227-7015(b)（2014 年 2 月）所述權利。

商標資訊

NETAPP、NETAPP 標誌及 <http://www.netapp.com/TM> 所列之標章均為 NetApp, Inc. 的商標。文中所涉及的所有其他公司或產品名稱，均為其各自所有者的商標，不得侵犯。